Leveraging Physical Locality to Integrate Smart Appliances in Non-Residential Buildings with Ultrasound and Bluetooth...

Conference Paper · April 2016

DOI: 10.1109/IoTDI.2015.35

CITATIONS
0

READS
82

4 authors, including:

Jonathan Fürst
IT University of Copenhagen
8 PUBLICATIONS **2** CITATIONS
SEE PROFILE

Philippe Bonnet
IT University of Copenhagen
94 PUBLICATIONS **2,239** CITATIONS
SEE PROFILE

Leveraging Physical Locality to Integrate Smart Appliances in Non-Residential Buildings with Ultrasound and Bluetooth Low Energy

Jonathan Fürst*, Kaifei Chen†, Mohammed Aljarrah* and Philippe Bonnet*
*IT University of Copenhagen, †UC Berkeley
jonf@itu.dk, kaifei@berkeley.edu, {moal, phbo}@itu.dk

***Abstract*—Smart appliances and sensors have become widely available. We are deploying them in our homes to manage the level of comfort, energy consumption or security. While such smart appliances are becoming an integral part of modern home automation systems, their integration into non-residential buildings is problematic. Indeed, smart appliance vendors rely on the assumption that the Local Area Network (LAN) guarantees locality and a single unit of use/administration. This assumption is not met in non-residential buildings, where the LAN infrastructure might cover one or several buildings, and where several organizations or functional units are co-located. Worse, directly coupling smart appliances to the Internet opens up a range of security issues as device owners have very little control over the way their smart appliances interact with external services. In order to address these problems, we propose a solution that couples the use and management of smart appliances with physical locality. Put differently, we propose that smart appliances can be accessed via smartphones, but only from the room they are located in. Our solution combines opportunistic connectivity through local Bluetooth Low Energy (BLE) with an ultrasound-based method for room level isolation. We describe and evaluate a prototype system, deployed in 25 offices and 2 common spaces of an office building. This work opens up intriguing avenues for new research focused on the representation and utilization of physical locality for decentralized building management.**

***Index Terms*—IoT; buildings; middleware; Bluetooth; BLE; ultrasound;**

I. Introduction

Since the late 1990s, researchers have postulated that sensors and actuators equipped with computation and communication capabilities would become widely available [1]. Today, this vision has become a reality. Environmental sensors and smart appliances such as thermostats, light bulbs, power plugs and locks equipped with short range radios are available in retail stores. A large part of these smart devices is targeted at home automation systems. However, there is a case to be made for the deployment of such smart infrastructure in non-residential buildings.

Buildings account for roughly 40% of primary energy consumption in the US and in Europe while we spend more than 90% of our time inside them [2, 3]. Larger non-residential buildings are massively instrumented and equipped with a Building Management System (BMS) under the control of Facility Management (FM).[1] Direct influence of occupants is often limited to few, physical switches (e.g., light switches). However, various studies show that personal comfort and energy consumption can be greatly improved by giving occupants direct personal control and by raising their awareness of their environment (see e.g., [4]). Likewise, existing BMS could be greatly improved by gaining more insight through additional sensors and by receiving direct feedback from building occupants [5]. Medium and small commercial buildings on the other hand are normally run without a BMS [6]. Consequently, the deployment and federating of a smart infrastructure could be a cost-effective way to introduce user centric management in such buildings.

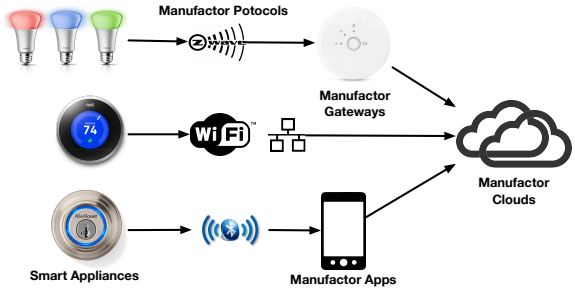


Fig. 1. Current Home Appliance IoT State

But the current IoT infrastructure does not align with the paradigms of non-residential buildings. First, despite being marketed the Internet of Things, the present ecosystem is characterized by manufacturer silos, with different protocols, different application gateways and different cloud infrastructures. Windley describes IoT as "The CompuServe of Things" [7]. We argue that this problem is even more severe when such devices are deployed in a business environment like a commercial building.

In principle, we can currently distinguish between three different IoT approaches to devices connectivity: (i) a low power protocol (e.g., ZigBee, ANT, Z-Wave) is integrated with the LAN infrastructure through a hardware manufacturer application gateway, (ii) the device is directly connected to the LAN (Ethernet/Wi-Fi), (iii) the device connects to a manufacturer application on the user's smartphone (see Figure 1).

A first consequence is that users do not have much control over appliances. Kaspersky Lab referred to IoT as the *"Inter-*

[1]The BMS centrally controls functions like HVAC or lighting.

net of Crappy Things" [8]. Appliances invisibly communicate with the manufacturer server, possibly data that the users might not want to share. The point is that with current IoT approaches, users have no control over the sharing of data which might be personal.

Second, the connectivity options presented above directly expose appliances to the LAN or the Internet. A potential attacker might thus take over control over appliances remotely, even when the owner is not present.

Third, the owner is responsible of keeping the software on all devices up to date, while she must rely on the manufacturer to provide timely updates for security flaws. Such a system would become unmanageable in a non-residential building.

Fourth, the setup, deployment and access patterns of current IoT devices are an ill fit for (semi-) public and shared spaces like non-residential buildings. After deployment in the LAN, the user needs to establish a bond (authorization) with an application—usually on the users' smartphone—and a cloud service by the smart device manufacturer. This introduces the following problems:

- In a residential domain, the LAN guarantees locality. In a non-residential building, the LAN might cover the whole building, or even several buildings. The notion of physical locality is thus lost.
- At home, there is a uniform user domain (residents and guests). In a non-residential building, the LAN covers multiple functional units within an organization, and possibly multiple organizations. Above described setup process does not allow to configure who has access to a given smart device in a non-residential building.
- While there might be social tensions within a family to control the home automation systems, a home constitutes a single unit of administration. In a non-residential building, facility management, the IT department and the organizations occupying a given building constitute multiple overlapping units of administrations.

Put differently, the integration of smart devices relies on the amalgamation of physical locality and unit of use/administration. This coupling is implicitly defined by the LAN domain at home, and explicitly by a password based authentication. There is no implicit coupling in non-residential buildings, and the explicit coupling is cumbersome to achieve.

In order to address these problems, we must thus:

1) decouple smart appliances from the LAN or Internet. We propose to abstract the silo-ed cloud communication from traditional IoT solutions behind a generic and opportunistic gateway device: the user's smartphone.
2) define an infrastructure that explicitly reconstitutes the coupling between physical locality and unit of use/administration. Our insight is therefore to leverage Bluetooth Low Energy together with ultrasound based data communication to achieve authorization at the granularity of a room within a non-residential building.

In this paper, we present the design and implementation of a decentralized system of smart devices (sensor and actuator nodes) that are only intermittently connected through smartphones equipped with Bluetooth Low Energy: *BLEoT–The Bluetooth Low Energy of Things*. Our solutions makes smart appliances directly accessible for all users within BLE-range via smartphone. Authentication and authorization is achieved through the transmission of a key from smart devices to nearby smartphones using sound signals above the human hearing range that work well with off-the-shelf smartphones. Our solution does not require any changes in the buildings' IT infrastructure and does not necessitate an extended authorization process. Instead, it requires a simple deployment procedure managed by facility management and relies on authentication/authorization that is based on physical locality.

We studied how our design enables the deployment of smart devices with a prototype system deployed in 25 offices and two common spaces. Our experiments show that our solution was easy to deploy, easy to manage and that it opens up intriguing avenues for new research focused on the representation and utilization of physical locality in the Internet of Things.

In summary, the main contributions of our work are:

- The design of a solution to the problem of deploying smart appliances in non-residential building, that combines (i) the use of sound signals above the hearing range to achieve a secure access of smart infrastructure for close-by smartphones, and (ii) opportunistic communication via user smartphones equipped with Bluetooth Low Energy, that serve as generic links between local smart infrastructure and the cloud in the context of IoT.
- The implementation and evaluation of such a system in the scope of 25 offices and two common spaces.

The remainder of this paper is structured as follows. First we present background and related work. We then describe the design principles we followed, and overall system architecture before moving on to the detailed presentation of the two core components of our system: Acoustic channel and Bluetooth-based opportunistic gateways. Finally we discuss implementation and evaluation of our deployment in a non-residential building.

II. Background and Related Work

The two central components of our solution to connect smart appliances in a non-residential building are (i) opportunistic communication via Bluetooth Low Energy (BLE) and (ii) ultrasound communication. In this section, we present related work on smart appliances connectivity. We give a short BLE primer and discuss the use of Bluetooth for opportunistic communications. Finally, we discuss existing work where ultrasound is used for managing locality and data transmission.

A. Smart Appliances Connectivity

There has been a number of approaches from industry and academia focused on connecting smart appliances to home automation systems. Google is leading a consortium to develop a networking protocol for IoT called 'Thread' [9]. Thread builds on 6LoWPAN and creates a mesh network of up to 250 devices. Thread is targeting the home market and provides IP

connectivity to all nodes. Apple recently introduced HomeKit, a framework for communicating with and controlling smart appliances [10]. HomeKit is built for the residential market, performing authentication based on the user's Apple ID. Our system distinguishes itself from these approaches by targeting non-residential buildings, where shared spaces require explicit coupling between use/administration domain and physical locality. We do not base our system on a fixed infrastructure like in Thread or Homekit, which does not scale to non-residential building, but on opportunistic gateways and self sustaining nodes.

The Californian start-up Zuli recently introduced BLE enabled smart-plugs, promoting a direct smart-plug to phone connectivity without the detour over a Wi-Fi network [11]. However their design follows a typical vertical vendor integration and smart plugs do not communicate their state with each other.

Zachariah et al. presented the idea of using mobile phones as routers for smart devices in [12]. They present the idea of using a participatory approach to bringing IPv6 to devices or to proxy their Bluetooth profiles to the Internet, but do not follow further with an implementation. With BLEoT, we have implemented a device bridge that creates Bluetooth profiles for REST APIs of several off-the-shelf smart appliances and opportunistic Internet gateways that enable communication between Bluetooth peripherals and between peripherals and Internet services.

Many prototypes and demonstrations in academia have been proposed for smart homes, including [13, 14, 15]. Mozer states in [14], that deployed smart systems need to inform users about their behavior. Functioning of devices need to be transparent to the user. Key challenges for home automation have further been addressed in [16], where Brush et al. list the following barriers: high cost of ownership, inflexibility, poor manageability, and difficulty achieving security. Brush also name convenience as a primary factor of user acceptance. With BLEoT, we use these observations as starting points for our design. Much work remains to be done to identify and address barriers to adoption in non-residential buildings.

B. Bluetooth Low Energy

We now describe briefly some of the key aspects and limitations of Bluetooth Low Energy (BLE) as they are important to understand our subsequent design decisions.

In BLE, data is exchanged asynchronously and limited to a single-hop in the 2.4 GHz band. BLE has two types of channels, advertising and data channels. If a device needs to only broadcast data, it can use the advertising channels to achieve a $1 : n$ unidirectional communication. Bidirectional communication takes place on the data channels between a central and a peripheral device, where peripherals usually provide services (server role) and centrals access these services (client role). Services are structured into characteristics that a client can read from or write to. This is managed by the Generic Attribute Profile (GATT). Following the publish-subscribe pattern, clients can get notified of value changes (notifications/indications). This can be used to push updated sensor values to clients for instance. The value that can be read and transmitted from a characteristic at a time is commonly limited to 20 bytes. Transmitting more than 20 bytes requires a split into multiple packages and possibly multiple read requests. A read/write request can only be initiated by a central device.

Park and Heidemann have shown that Bluetooth can be efficiently used as a support for data muling. During their deployment in several environments, they note that office spaces are a specially good fit for opportunistic mobility due to their dense sensors and long human loiter times [17]. They base their system on Bluetooth 2.0, using small embedded PCs and USB dongles. In contrast, our system utilizes custom battery powered sensor nodes, as well as commodity appliances and smartphones that communicate using Bluetooth Low Energy.

The recently announced Bluetooth 4.2 has the goal of establishing BLE as the wireless standard for IoT. It introduces a new profile enabling IPv6 for Bluetooth [18]. Nordic Semiconductor reacted by providing IPv6 over a Bluetooth protocol stack as well as a prototype of a IPv6 router, implemented on a Raspberry Pi [19]. Our implementation is not based on IPv6, instead we are proxying the existing APIs of smart appliances to native Bluetooth services.

C. Sound for Locality and Data Transmission

Sound signals have been used to achieve both, (i) localization/proximity and (ii) data communication.

Madhavapeddy et al. emphasize the adequacy of sound as a means to manage localization. Especially in buildings, walls prevent audio signals to be propagate outside a room, enabling room level localization [20]. Priyantha et al. have established the use of concurrent radio and sound signals to infer distance [21]. Borriello et al. then uses a combination of sound modulation and Wi-Fi networking (as communication channel) to achieve room level localization [22]. Finally, Lazik and Rowe use chirps in the ultrasound range to achieve localization [23].

Sound modulation for data communication has been studied extensively. Madhavapeddy et al. give an overview of using acoustic communication for both long range (telephone line) and short range (3m). For inaudible sounds (OOK on 21.2kHz carrier), they achieve 8bps [24]. Gerasimov and Bender explore different data encoding schemas for acoustic data transmission (echo coding, PSK, FSK and impulse coding) both in audible (5.5 kHz) and inaudible (18.4 kHz) frequency ranges. Their implementation works well for transmitter-receiver distances of up to 2m. Their maximum data rate is 3.4 Kbps when using multiple-level B-FSK in the 18.4 kHz range [25]. Nandakumar et al. propose a system to replace NFC communication with a secure, short range, sound based protocol that works in a range up to 20cm. Their operating bandwidth is 1kHz in the range of 6-7kHz. Their system relies on orthogonal frequency division multiplex (OFDM) with binary and quadrature PSK modulation for digital modulation and achieves 2.4Kbps [26]. Lee et al. develop an aerial

acoustic communication by adopting chirp signals for digital modulation [27]. Their system works from 19.5-22kHz and achieves 16bps on a range up to 25m, while relying on a backend server to overcome the low data rate. Lopes and Aguiar develop a modulation schema that works in audible frequencies, but is more pleasant for humans by emulating sounds that humans are used to, like the ones of birds. They mention a data rate of 100-1000bps, but do not discuss the maximum range of their design [28].

In BLEoT, we are building on the experiences developed in aforementioned work. We achieve room locality through sound signals in the ultrasound range. We also use these signals to transmit an alternating key to close-by smartphones. We thus directly couple device access with physical access.

III. Design Principles

In this section, we discuss the principles that underlie the design of the system that we call BLEoT. These principles are articulated around three key requirements when deploying smart appliances in a non-residential building: security, ease of use and decentralized control.

A. Security Model

Our security goals are twofold. First, we want to ensure that only valid users are able to access smart devices. Second, we want to limit the possibility of a remote attack on the smart infrastructure. The assumption in our security model is that we trust the smartphone devices.

To ensure that only valid users can access devices, we couple their authorization with physical locality. The local coupling of acoustic data transmission and authorization maps thus physical authorization with digital authorization. Put differently, if a person has physical access to a room, we assume that she also has access to devices in that room. This is analogous to a person being able to switch the light in a room using a physical switch. This assumption is consistent with the security model of the physical space.

B. Deployment

Heavy deployment overhead has been one of the major challenges for smart systems and home automation in the past [16]. A key motivation of our work is to enable a quick and possibly temporary deployment of sensors and smart appliances in a large number of shared spaces. Ideally, neither facility management nor the IT department are involved in the deployment of smart appliances. Users should be able to (i) deploy smart appliances and make them available, and (ii) access any smart appliances they get close to.

A key aspect of our system is the definition of a common interface that abstracts existing devices. Devices are only able to communicate via that interface through the user's smartphone.

C. Decentralized Control

Deployed smart appliances and sensors need to follow a decentralized control logic, where operation does not break down with a connection loss to a central control unit.

Traditional Building Management Systems are usually following a three-tier model of management, automation and field level. At the lowest level, sensors and actuators communicate through a field bus (digital serial data bus) with each other and with control devices of the upper automation layer. Communication at the automation and management level usually takes place over LAN. Automation can happen locally via a direct coupling of sensors and actuators (occupancy and light), on the automation level (via direct digital controllers (DDCs)) or on the management level (building automation control computer) [29].

These layers of communication and control enable a building to be operated even when communication between local controllers and the central BMS computer breaks down. E.g., a direct coupling of a temperature sensor and a thermostat will be operational even in the absence of the centralized BMS. This approach to fault tolerance has been proven successful during several decades of building automation. A system of smart appliances and sensors should also be based on local control capabilities that do not require permanent connectivity with a central server.

Our design, based on opportunistic connectivity, pushes this logic and reverses the assumption: sensors and actuators are not connected to the upper automation layer unless a user with a smartphone equipped with the appropriate app enters the room where they are located.

IV. Design Overview

BLEoT aims to provide secure and usable access to smart appliances in non-residential buildings. BLEoT replaces existing manufacturer stovepipes with a gateway design. Edge-to-cloud data exchange becomes more visible and controllable by the user, via their smartphone.

A. Architecture

BLEoT consists of a loosely coupled three-tier architecture that uses opportunistic gateways to connect IoT devices with each other and with the cloud (see Figure 2). Locally, nodes act as BLE bridges for sensors and smart appliances. They advertise their service and state periodically via BLE. Nodes that bridge actuators contain a second, acoustic communication channel in the form of ultrasound. This channel is used to transmit a periodically changing key (e.g., every hour) that is required to access actuator capabilities on a node (see Figure 3). As such, we achieve a room level authorization based on the attenuation of sound waves by walls and doors. Nodes do not directly connect to each other or to remote services (e.g., the manufacturer cloud web service), but rely on opportunistic gateways—in form of smartphones. Using their BLE-enabled smartphone running BLEoT as gateways, users can interact with the environment (e.g., switching the light) in a room they enter and to read local sensor information (e.g., temperature).

All nodes are equipped with a Bluetooth radio, and some computation and storage capabilities. Nodes that connect to actuators are also equipped with a speaker. Nodes can connect

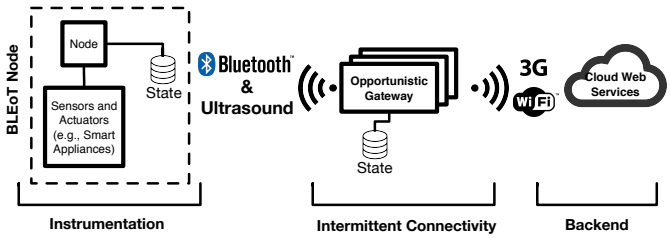


Fig. 2. Overall Architecture of BLEoT

to any sensor and actuator. We have for instance implemented several prototypes of native sensor and actuator nodes (see §VII). However, most importantly, our design allows to integrate various off-the-shelf smart appliances by bridging them from their native technologies and protocols to BLE services.

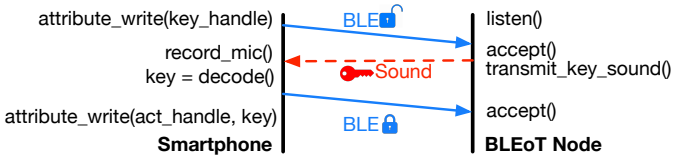


Fig. 3. Sound Based Authorization Process

V. BLE-Based Smartphone: A Gateway to Rule them All

This section discusses the design of our BLE based opportunistic gateway infrastructure. We describe how we abstract off-the shelf devices to a BLE interface and how we define the protocol that allows node-to-node and node-to-cloud communication.

A. Bridging IoT Devices

We abstract off-the-shelf smart appliances through a bridge to a BLEoT Node. Current off-the-shelf smart appliances provide different technologies, physical layers and APIs for access and are vertically integrated. Further, they do not always provide open and documented APIs.

Luckily, this problem of integrating several incompatible technologies, physical layers and (closed) APIs has been recognized and investigated earlier. Research projects as well as efforts originating in the open source community deal with their integration (e.g., [30, 31, 32]). BLEoT is building on the domain knowledge and technical insight gained in previous work (namely sMAP [30] and FHEM [31]). Both projects provide a rich repository of drivers for several existing IoT devices and the capability to write new ones. Such a device driver implements the specific protocol of the device and maps the device's functions to a local REST interface. Finally, this REST interface can then be accessed by applications independent of the device manufacturer interface (e.g., openHAB provides a UI application for residential homes [32]).

With BLEoT, we extend the current practice of REST integration by adding the capability of making that interface available locally via BLE to clients in proximity.

TABLE I
BLEoT Local Sensor Service

Characteristic	Access	Example	Description
Requested value	Write	`ed0000ff`	Value of requested local sensor value.

Taking the limitations of BLE into consideration, we have implemented two different approaches: (i) We simply tunnel the REST interface though BLE and (ii) we create native BLE services for each smart appliance.

While the latter approach (ii) seems like the better choice – because it is free of the HTTP overhead–, it causes difficulties when the set of smart applications connected to a bridge evolves in time. The commonly used Bluegiga BLED112 dongle requires for instance a reprogramming via USB DFU using a proprietary Windows-only update utility that writes a licence key on the chip. So we prefer the former approach (i). By tunneling the rest interface over a RX/TX pipe which we implement as a service on the BLE dongle, we achieve a generic interface. Standardized metadata that exists on the local REST interfaces becomes available on the smartphone client.

B. Common Interface

The interface between nodes and gateways consists of data communication taking place in pure advertisement state as well as communication taking place when a gateway is connected to a node.

1) Gateway Services: Each node exports BLEoT gateway services. Each service has a number of BLE read and write characteristics. Gateways access these services when a node forwards a request through its advertising message.

Table I shows the our service that allows basic node-to-node state transfer. A peripheral requests a sensor value from a defined set of abstracted types (e.g., temperature, humidity, CO_2 etc.). Opportunistic gateways serving that request will then directly write that value to the *Requested Value* characteristic.

Table II shows these characteristics for the BLEoT HTTP Service, while demonstrating a typical data offloading task that we implemented in our deployment. A URL characteristic provides the resource in form of a URL, while the HTTP method and payload are provided by other characteristics. Our specific payload for offloading sensor data consists of the sensor value (using IEEE 754 floats) and the sequence number of the measurement. Two writeable characteristics allow opportunistic gateways to acknowledge success in form of HTTP response codes and possibly write back the result in form of a HTTP message.

2) Advertisement Protocol: BLEoT nodes advertise their state, as well as requests for gateway services, regularly using a defined format on the advertising channels. This allows a $1 : n$ unidirectional communication between a node and close-by gateways. The data structure of the BLEoT advertisement can be seen in Figure 4. All data, necessary to populate a User Interface (UI) on the smartphone is encoded in the payload

TABLE II
BLEoT HTTP SERVICE

Characteristic	Access	Example	Description
URL	Read	130.226.142. 195/api/bleot/ addreadings	URL for the HTTP request.
HTTP method	Read	POST	HTTP method to use.
Payload	Read	`ed0000ff81ff ... 81ff0100`	Payload for the HTTP request.
HTTP response	Write	201	HTTP response codes.
Message body	Write	not used	HTTP message (e.g., new configuration).

together with state and service request information for the gateway service:

- The **Manufacturer ID** allows to distinguish BLEoT Nodes from other BLE devices.
- The **Human Readable Location** enables a direct display of the node's location in a UI without requiring any remote connection. This makes each of our smartphone applications usable on every building.
- The **Service Request** allows a node to request of gateways to perform a HTTP service or local service in its behalf.
- A **Misc** flag encodes several information in binary: battery level, sensor/actuator and the expected length of a service request.
- The current **State** of the node is encoded as a IEEE floating point number (e.g., current sensor value).
- The **Coordinates** flag abstracts common placements of nodes in a room (e.g., `ceiling`, `outside` etc.).
- **ID** gives a node a unique ID inside a building scope.

The Bluetooth standard restricts BLE advertisement packets to 24 Bytes. This limits the maximum length of the dynamic, human readable location string to 11 Bytes in our protocol. But it is sufficient for representing a typical "building, room number encoding" (e.g., `ITU4D21`, `SODA410` to the user.

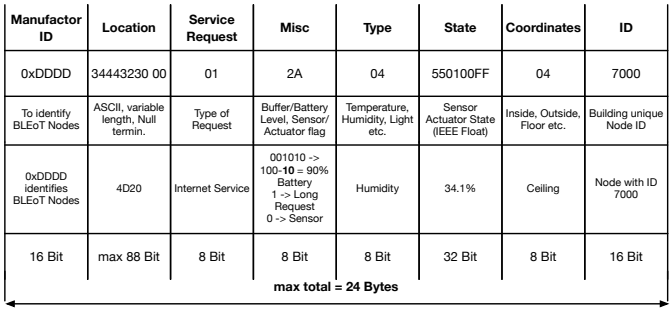


Fig. 4. BLEoT payload

The example payload in Figure 4 shows how we use the protocol described above for one-to-many communication. The depicted node is located in the ceiling of room "4D20". The location is ASCII encoded, which allows direct display in a UI application. The node is currently requesting Internet services from gateways, its battery level is at 90% and the service request it has will take relatively long time for the gateway to complete. We have divided service requests in long and short lasting requests. Currently, only data-offloading is a long lasting request. This allows a gateway to utilize its inbuilt location service to decide if it should accept such a request (person is sitting at his/her office and phone does not move) or not (person is walking the hallway). Lastly, the exampled node is advertising a humidity sensor value of 34.1%.

C. Deployment Processes and Configuration

We expect that BLEoT Nodes will be mostly deployed by building users. Hence we can not expect that deployment requires a technical background with a deep understanding of Computer Science or the operation of a BMS. Our current implementation of the deployment and configuration process reflects this:

leftmargin=*

1) A user places the BLEoT node at the deployment location and pushes the configuration button on the node.
2) In the configuration tab of our smartphone app, the node is now shown ready to be configured. When connecting to the node we establish a bond, storing a long term AES-CCM 128bit key on the node and the smartphone. This key is necessary when a node needs to be reconfigured.
3) The user defines location (building, floor no. and room no.) and further narrows it down by choosing from predefined coordinates associated with that location (ceiling, floor, outside...). She then configures each single sensor and actuator of that node. For sensors, she can enable data-offloading, set sample, advertising frequency, buffer threshold and signal strength or leave them at their default. For actuators, she is able to define a generic control in the form of a default schedule for the space. In practice these are things like defining actuator set points for daytime and night-time, for weekdays and weekends. If there is no user in the space, then these settings will define the behavior of the space. To adjust the acoustic channel to the size of the space, we implement an adaption procedure, where the user moves at the room border and sound volume is adjusted to the quality of the received signal—now the node is operational.

The setup of off-the-shelf, smart appliances is slightly more complex. Before the above process, a BLEoT bridge is configured that connects all these devices and forms a BLE access point in form of a BLEoT node. Only then can the node be configured as usual.

Our design allows for an update of node configurations through opportunistic gateways. The node advertises on a regular basis its request (in the form of a HTTP service request) to get a possible new configuration. The gateway gets the configuration from the provided remote server and writes it back to the node. The message is encrypted server side using

the AES-CCM 128bit keys that have been generated during the first bond with the smartphone of a trusted deployment person.

VI. Acoustic Channel

The acoustic channel between smartphone devices and BLEoT nodes implements room level authorization. As a requirement, our system must be capable to operate in indoor, (semi-) public spaces. BLEoT's correct operation depends on the acoustic channel as a physical medium. First, indoor spaces contain some level of ambient noise that causes interference to any acoustic communication. Second, indoor environments cause sound signals to echo from walls and other obstacles. The resulting multipath propagation represents another challenge. Finally, our requirement is to operate above the human hearing range and with off-the-shelf devices (smartphones).

A. Ambient Noise

Ambient noise differs on the type of indoor location. We categorize spaces into (i) smaller, delimited spaces like offices, (ii) spaces in which people pass through, like hallways and (iii) spaces of gathering (meeting rooms, cafeteria), where the ambient noise is expected to be higher. To qualify their noise level, we have taken sound samples on different periods of the day.

As our requirement is to work above the human hearing range, we pay special attention to ambient noise in higher frequencies. These might be due to the influence of high frequency sounds like those commonly emitted by switched-mode-power supplies. In Figure 5 we plot the Power Spectral Density (PSD) for different locations on our campus. The data was recorded using a portable USB condenser Microphone (Samson Go Mic). As can be seen, regardless of the environment, high spectral power can especially be observed up to around 6000kHz. High frequencies are not critically affected by ambient noise.

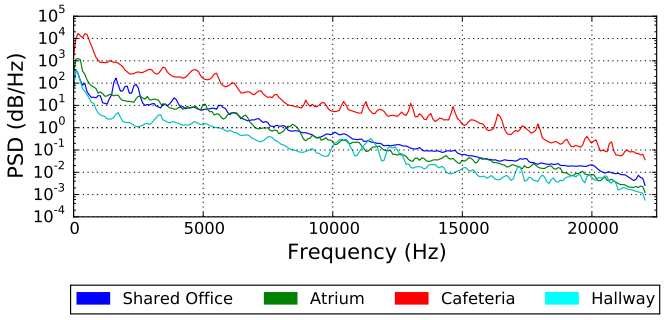


Fig. 5. Indoor Ambient Noise

B. Frequency Selectiveness

The human hearing range differs from person to person and depends on the applied audio pressure. Previous studies come to different results, reaching from a maximum of 19-20kHz ([33]) to 18kHz ([34]). Modern smartphones usually have an acoustic sampling rate of up to 44.1kHz, leading to a maximum frequency of 22kHz in theory. However, smartphone microphones are usually quite frequency selective and optimized for human voice frequencies (see e.g., [26]). We conducted experiments with several commonly used smartphones by playing a wide, linear chirp over the full frequency range (up to 22kHz). To minimize side effects of a non-flat speaker frequency responses, we used a studio monitor speaker (Audioengine 5+). Figure 6 shows our result for calculating PSD after Welchs method for several smartphones and tablets. As can be seen, the frequency selectiveness depends on device type and frequency. Higher frequencies contain a wider variance, but most devices are capable to record up to 21-22kHz. The result corresponds to the results obtained in [23] and [27].

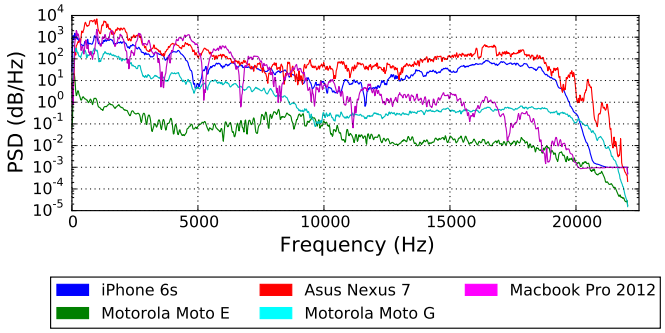


Fig. 6. Frequency Selectiveness

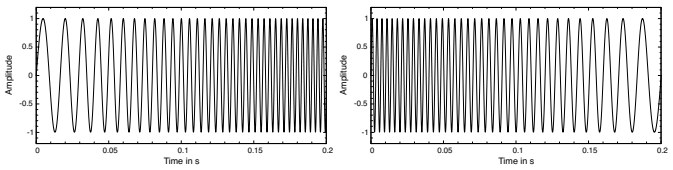
C. Generating the Signal

Our ambient noise experiments (see VI-A) have shown that high frequency ranges show little to no disturbance by ambient sounds. The sensitivity of smartphone microphones show however a high variance between models (see VI-B). Due to the high frequency selectiveness of smartphone microphones, we disregard any Frequency Shift Keying (FSK) based modulation schemas. Phase Shift Keying (PSK) schemas on the other hand perform bad in time-varying, fading channels like the acoustic channel [35]. We therefore follow the insight gained in [23] and [27] to use chirp signals for binary acoustic communication.

Compared to the time-invariance of FSK and PSK, a chirp signal varies its frequency over time. A chirp is a signal that linearly increases (up-chirp) or decreases (down-chirp) its frequency between two frequency ranges (see Figure 7). Chirp signals have been used extensively in sonar and radar applications. In the context of frequency selective microphones, a chirp signal has the advantage to use the same frequency range for up- and down-chirps. This means that both chirps are affected in a symmetric way.

Linear, up- and down-chirps are defined as follows:

$$s_{1,2}(t) = \sin\left[\phi_0 + 2\pi\left(f_0 t + \frac{f_1 - f_0}{2T}t^2\right)\right] \qquad (1)$$



(a) Up-Chirp (b) Down-Chirp

Fig. 7. Up- and down-chirp signals between 50 and 300Hz and over a time interval of 0.2s.

Where:

- f_0 is the starting frequency.
- f_1 is the final frequency.
- T is symbol duration.
- ϕ_0 is the initial signal phase at $t = 0$.

Applying Equation 1 directly for modulating the signal results in a human perceivable clicking noise at the begin and end of the signal. This is due to the instant shift into high frequency ranges [23]. To remove this clicking effect we simply apply each chirp signal with a Hann window. Error detection is achieved by transmitting a parity bit at the end of the message.

To avoid multipath interferences, we add 50ms guard interval between symbols. We set our symbol duration to 100ms as our system is not expected to transmit longer messages via sound, but a small secret to allow clients to authenticate themselves with the BLEoT node. This results in a data rate of 7bps.

Acoustic power decreases by the square of the distance from the transmitter. This makes a one-fits-all power setting difficult. Ideally, the signal power changes with the receiver distance in a room. We therefore make use of the Bluetooth channel to achieve a more adaptive sound transmission by using RSSI at the transmitter as a coarse estimator.

RSSI roughly relates to distance as follows [36]:

$$RSSI = -10n \times log_{10}(d) - A \tag{2}$$

Where:

- n is the signal propagation constant.
- A is received signal strength at 1m distance (dBm).
- d is the distance between sender and receiver (m).

We experimentally define n and A for our combination of radios and indoor environment by taking RSSI measurements every meter from 1-25m. With $A = -59.947$, we then calculate n by inserting A in Equation 2 using the values from each experiment sample. This resulted in an averaged value of $n = 2.772$.

In the transmitter, we then multiply the signal power with the squared distance estimation.

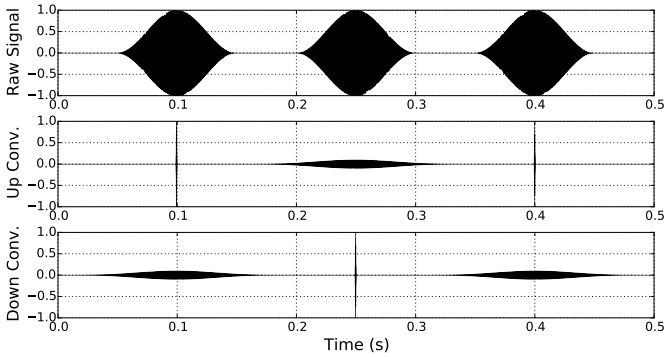


Fig. 8. Decoding `101` signal by convolving with the time reversed chirps.

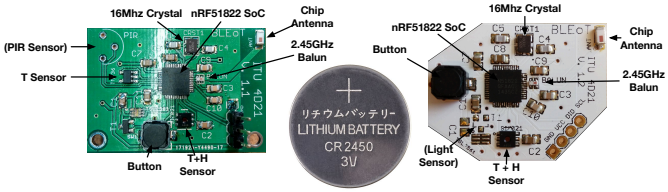


Fig. 9. BLEoT Sensor Nodes

D. Receiving the Signal

To decode the signal at the receiver, we apply a combination of convolution and peak detection. We generate both, up- and down-chirp signals, time reverse them and convolve them with the received audio signal. We pad the signal with 0s to achieve a complexity of $N * logN$ for FFT. After converting back to the time domain of the signal, we then perform peak detection on both results (received signal convolved with up-chirp and down-chirp). By merging both sets of detected peaks, we can then sort the values by time and thus decode the message.

Figure 8 shows a signal of `101` and the result of its convolution with the reversed chirp signals.

VII. System Implementation

We design and build custom nodes that connect sensors and actuators as well as a node that simply provide a Bluetooth bridge for off-the-shelf smart appliances.

The custom BLEoT nodes are implemented using the October 2014 revision of Nordic Semiconductor's nRF51822 SoC. It features a 32-bit ARM Cortex M0, 256kB of flash storage (of which the Bluetooth stack requires 80kB), 32kB of RAM and a 31-pin GPIO mapping scheme for analog and digital in- and output. We have designed and implemented several types of sensor (motion, temperature, humidity, light) and actuator (AC relay with Hall effect current sensor) boards using the nRF51822 (Figure 9 show our sensor nodes). The bulk of sensor nodes runs from a single 3V coin cell battery of type CR2450 which comes with a capacity of 620mAh.

Our IoT bridge is implemented using a Raspberry Pi Model B with a Bluegiga BLED112 dongle and a single, active speaker. The programmable dongle makes it a connectable peripheral device. We have experimented with several smart

appliances. Our current deployment relies on Philips Hue light bulbs as personal desk lights, HomeMatic radiator thermostats to allow thermal changes and HomeMatic for switching arbitrary appliances. The Philips Hue lights rely on Z-Wave as the communication protocol, while the HomeMatic appliances use a proprietary protocol on the 868MHz band (BidCos).

A. Gateways

We implement opportunistic gateways in the form of an Android background service that is bundled with a UI application allowing users to control and sense their local space.

Once the application is started, the background service bootstraps necessary functionalities and maintains a global state. It relies on a fixed thread pool executor that will drive a number of internal threads concurrently. A "packet interceptor" thread is cyclically scanning for BLE advertisement packets, which then are handled by a "packet handler" thread that notifies observers (the UI) in case a valid BLEoT packet with new data is available. If the packet contains a service request flag, the thread will create a special "Service-Thread" and add it to the pool. This thread will indicate its accept of the service request by establishing a connection to the node and initiating the demanded request (e.g., data offloading its historic values). A "packet cleaner" thread is further removing dated packets (currently after 90s) and notifying the observers. This makes sure that possible occupant movement is reflected in the UI.

B. Cloud Web Services

We implement several Web services that nodes can access via opportunistic gateways: (i) data-offloading that allows nodes to offload their buffered sensor data, (ii) configuration retrieval to update a node's configuration. Both are implemented in Go (https://golang.org). The off-loading service takes POST requests that contain as their payload a node's buffered sensor data. Sensor data includes a sequence number for each sample that enables the logic on server side to calculate time stamps backwards, based on the known sample frequency. We plot historical sample data and make it accessible for other applications. The configuration retrieval service provides a new configuration for the node after a GET request is issued. Both services are protected by per node unique symmetric keys that are created when a node is deployed for the first time.

VIII. Results

We deployed our system in 25 (mostly shared) offices, two hallways/meeting rooms and a kitchen area at our university. The facilities consist of a long hallway with offices to the left and right. Meeting rooms are integrated in the hallway itself (see Figure 10).

We conducted experiments with BLEoT, as described in §VII, with various Android devices. Performance results varied. In the following, we show the results obtained with two devices that illustrate the breadth of the performance spectrum:

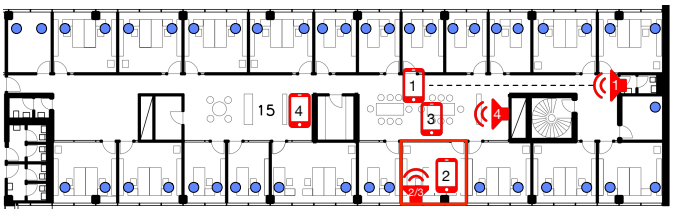


Fig. 10. Deployment with BLEoT nodes (circles) and experiments setup

(i) the Motorola Moto E smartphone and (ii) the Google Nexus 7 tablet.[2]

The most important performance measures are latency (for acoustic transmission and BLE), the performance of opportunistic gateway services (we evaluate data offloading as an example service), energy consumption (on battery powered nodes and smartphones) and finally the capability of our design to achieve a sound based room isolation. We conclude with a discussion on security aspects and a qualitative placement and discussion of our work.

A. BLE Channel

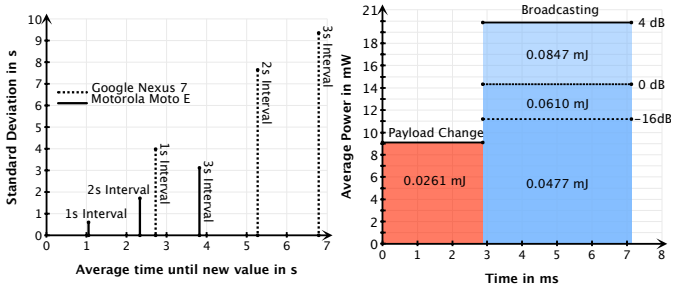
1) Latency: Several latency metrics are relevant for our work. Latency of actuation and the retrieval of local sensor values (state) is important to end-users (humans are able to perceive switching delays greater than 100ms [37]). The latency of opportunistic gateway services is ultimately important for the working of our system: If the state of a node drifts too far away from the state of its environment, its data might become irrelevant; if it is not able to offload its data, then data might be lost.

Our experiments have shown that latency of state transmission through advertisements depends heavily on the specific smartphone and on the advertising interval that has been chosen on the nodes. Figure 11a shows a growing latency as a result of a growing advertising interval. The standard deviation for the Nexus 7 is much higher than the Moto E.[3] Due to these erratic results between different devices, we suggest to set the interval to a maximum of 3s, which in our tests has been sufficient to keep the maximum latency between advertisements below 8s while still preserving battery.

Further, we measured the experienced latency when actuating bridged, off-the-shelf smart appliances. It took on average 0.6s to discover a new BLEoT node (with advertising interval set to 30ms). It then takes 3.74s to receive a 16bit key via the acoustic channel and the services of the node. Now, actuation can be done relatively quick. We measured the overhead introduced by our system in the range of 70 to 130ms. Node discovery and service retrieval only needs to be performed when a user enters a space for the first time. Afterwards, actuation occurs timely.

[2]See http://www.motorola.com/us/smartphones/moto-e-2nd-gen/moto-e-2nd-gen.html and http://en.wikipedia.org/wiki/Nexus_7_(2013_version)

[3]The unpredictability of the Nexus 7 seems to be a known problem in the Android community without a fix (https://code.google.com/p/android/issues/detail?id=65863).



(a) Receiver Latency (b) Energy Consumption

Fig. 11. Energy Consumption and Receiver Latency for Advertising.

The latency introduced by a opportunistic gateway (node-to-cloud and node-to-node) depends mainly on its availability. We have not yet performed an evaluation with an extended user base ($n > 10$). However we argue that the need for a gateway is highest when people are actually occupying a space. If a space is not used, it can be run in a default, unoccupied schedule. Data offloading can become critical during times of holidays. The maximum available flash storage of 156kB on our implemented sensor nodes allows for the storage of 194500 sensor values (8 bytes each). A node with a single sensor and 1 minute sample period will be able to persist values over a period of more than 16 days. This is in many cases sufficient for longer unoccupied periods.

2) Data Offloading: Data offloading is a data intensive service. We conducted experiments both, right next to the offloaded sensor node and at a distance of approx. 20m. 3258 sensor values were offloaded for 10 re-runs, taking 82msec/value on average for the close offloading and 100msec/value on average for the 20m distant sensor node. Moving the gateway further away from the sensor node resulted in connection dropouts.

These results fit well with our deployment in an office space in a university building. It takes around 5 minutes to continuously offload 3258 values. This seems much, but it does not worsen general operation due to the long loiter times in office spaces. When deploying our system in another context, buffer threshold needs to be adjusted to the specific environment and its occupation model.

B. Energy

We evaluate energy consumption for our battery powered sensor nodes and for the opportunistic gateway services. Energy consumption is mostly relevant for BLEoT sensor nodes that run from battery. Nodes that bridge off-the-shelf smart appliances are connected to the mains.

Our test equipment consists of a digital oscilloscope (Rigol DS1054Z) that we combine with an op-amp circuit to amplify a voltage signal at a small burden resistor. Due to the spread in consumption of BLE (from a few μA during sleep to $15mA$ during peak current when transmitting), we are using different burden resistor sizes (1Ω for higher currents and 1000Ω for sleep currents). Together with our amplification circuit of factor 100, this gives us a range of $0 - 20$mA and $0 - 20\mu$A for respective resistors.

We have measured consumption for the main operational events of our nodes. Figure 11b shows averaged results of 100 broadcasting and payload changes. The energy spent on payload changes is independent on TX power. Energy consumption for sensing is mostly defined by the specific sensor type and we thus omit it here. However persisting a value to flash is independent of the chosen sensor. We have measured that storing a single measurement requires 5.9μJ. An important event for our opportunistic design is node data collection by a gateway. In such a case nodes are transferring relatively big amounts of data in chunks of 22 bytes BLE data packets. A single connection event takes on average 3.05s with a consumption of 1.11mJ/s. This adds up to a total consumption of 3.39mJ for establishing a connection. The consumption for offloading the data depends on how full the buffer is. Our offloading mechanism uses Bluetooth indications, which allow GATT servers (peripherals) to push new values to the client (central device) without the need of poll requests by the client. We have measured 0.21mJ per indication. As one data packet in an indication fits two of our measurement data structures (8 bytes each), we end up with around 0.1mJ per transmitted measurement. For transmitting n measurements the consumption model thus becomes: $3.39\text{mJ} + n \cdot 0.1\text{mJ}$. A sensor node with a single sensor, a sample frequency of 1min and advertising frequency of 3s will thus run well over a year from a single CR2450 battery.

To evaluate the impact on smartphone energy consumption, we use "GSam Battery Monitor". We run our gateway service for a period of one week on the Motorola Moto E. This resulted in a battery impact of 14.8% on the phone's battery. During this time, the phone was offloading 68424 sensor values to the backend. We expect that the battery impact of such applications will drop in the near future, as Bluetooth connected devices become ubiquitous and hardware and OS support is being optimized for it. Further, to incentivize the use of our gateway service, we allow users to decide if the service should only be active when the UI application is in forefront. Our experience has shown that the power consumption is then mainly determined by the display.

C. Acoustic Channel

We now look at the performance characteristics of the acoustic channel.

1) PRR and Distance: To evaluate the Packet Reception Ratio (PRR), we transmit a message of 52bits from different distances (every 1m, from 1 to 10m) at different locations. Figure 12 shows the result doing this experiment in our section hallway (see Figure 10, No. 1) an open space (our university's atrium) and a shared office (No. 2). In open space we reach 30m before PRR drops substantially due to signal overlappings introduced by multipath (see Figure 13). In the hallway, these overlappings occur much earlier (around 16m). We can cover the $18.5m^2$ office fully with our signal. Extending the range is possible by increasing the guard interval between signals.

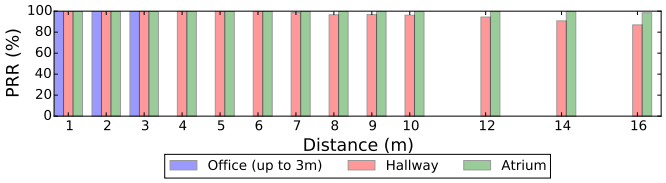


Fig. 12. PRR for different locations and as a function of distance.

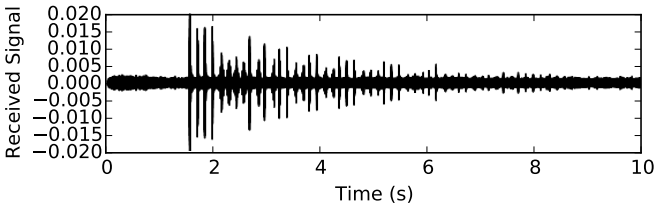


Fig. 13. Multipath effect on a signal at 40m distance.

Initial experiments when doubling the interval show that our approach can reach $> 40m$, which is more than the authors in [27] have achieved.

2) Room Isolation and Multipath: The main purpose in our design of using an acoustic transmission channel is to achieve isolation which we then map to device authorization. To measure the effectiveness of this isolation, we perform experiments with different pathways between transmitter and receiver (see Figure 10: No. 2, 3 and 4). Our experiments show that the acoustic channel easily covers the whole room, while a device listening at the wall and door, outside of the room is not able to pick up the signal. We successfully tested room isolation with different materials commonly used in buildings (concrete, wood, glass and polymer). An open door however, makes the signal available. To minimize this effect node configuration can be introduced to adjust the signal power to the room size. An attacker with a device with a high gain amplifier is however still able to pick up the signal.

D. Security Analysis

We use the concept of physical locality to authorize local actuation and access to sensor data. By basing access on locality in the physical world, we move the challenge of achieving a secure system out of a pure software implementation and into the physical security of a space. Security is therefore mainly dependent on how access to that space is controlled. Acoustic waves do not stop at open doors or windows. This means that our system further depends on the structure and location of a space.[4] Security also depends on the social structure that is prevalent. Are occupants likely to fiddle with other's instrumentation? During our deployment, one individual was occasionally actuating other's instrumentation as a friendly joke. Studying such scenarios in the context of a larger deployment is future work.

[4]E.g., A ground office is different than an office on the top floor.

We use symmetric encryption to exchange data between nodes and Web services, via smartphone gateways. Keys are created and exchanged when a node is deployed. During the deployment phase, local sniffing attacks might be conducted. After keys are exchanged, we depend on the security of the AES encryption. Mitigating the attack window during deployment is difficult. Nodes would need to have extended human input capabilities to enter a code directly on the node. Because of the short attack window and the exclusively local scope of the attack, we consider it a manageable threat.

IX. Conclusion

The BLEoT infrastructure is a first step towards integrating smart appliances in the context of non-residential buildings. Using ultrasound as a means to establish physical locality, BLEoT makes it possible to deploy and access smart appliances within non-residential buildings. Our results show that this approach is technically viable. Much work remains to be done to explore how BLEoT could complement an existing Building Management System (and possibly replace BMS in small buildings). More specifically, the key issue left as future work is the study of centralized building operation based on instrumented spaces that are only available when occupied by users with smartphone gateways.

In future work, we want to install our BLEoT sensors around the campus to perform larger user tests, to experiment with the acceptance of our system and to measure actual data loss due to the uncertainty in the system. We further want to experiment with the (temporary) deployment of sensors to improve the actual BMS of our campus.

Another interesting research area is to study how people will behave when they have the possibility to control smart appliances via smartphones. Having a larger installation, we will be able to evaluate to which extend an adaptive environment actually helps to increase comfort and reduce energy consumption. We want to compare an adaptive, user controlled setting with a central, model based system in terms of consumption and comfort.

References

[1] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next century challenges: Scalable coordination in sensor networks," in *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*. ACM, 1999, pp. 263–270.

[2] L. Pérez-Lombard, J. Ortiz, and C. Pout, "A review on buildings energy consumption information," *Energy and buildings*, vol. 40, no. 3, pp. 394–398, 2008.

[3] U.S. EPA/Office of Air and Radiation and Consumer Product Safety Commission, "The inside story: A guide to indoor air quality,," 1988.

[4] G. Brager, G. Paliaga, and R. De Dear, "Operable windows, personal control and occupant comfort." *Center for the Built Environment*, 2004.

[5] A. Piette, K. Kinney, and P. Haves, "Analysis of an information monitoring and diagnostic system to improve building operations," *Energy and Buildings*, vol. 33, no. 8, 2001.

[6] S. Katipamula, R. M. Underhill, J. K. Goddard, D. Taasevigen, M. Piette, J. Granderson, R. E. Brown, S. M. Lanzisera, and

T. Kuruganti, "Small-and medium-sized commercial building monitoring and controls needs: A scoping study," *PNNL, Richland, WA (US), Tech. Rep*, 2012.
[7] P. Windley, "The CompuServe of Things," http://www.windley.com/archives/2014/04/the_compuserve_of_things.shtml, 2014.
[8] Kaspersky Lab, "Internet of crappy things," https://blog.kaspersky.com/internet-of-crappy-things/7667/, 2015.
[9] Thread Group, http://threadgroup.org, 2015.
[10] Apple HomeKit, https://developer.apple.com/homekit/, 2005.
[11] Zuli Smartplug, https://zuli.io, 2015.
[12] T. Zachariah, N. Klugman, B. Campbell, J. Adkins, N. Jackson, and P. Dutta, "The internet of things has a gateway problem," in *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications*. ACM, 2015, pp. 27–32.
[13] S. S. Intille, "Designing a home of the future," *IEEE pervasive computing*, vol. 1, no. 2, pp. 76–82, 2002.
[14] M. Mozer, "Lessons from an adaptive house," Ph.D. dissertation, Architectural Engineering, 2004.
[15] C. D. Kidd, R. Orr, G. D. Abowd, C. G. Atkeson, I. A. Essa, B. MacIntyre, E. Mynatt, T. E. Starner, and W. Newstetter, "The aware home: A living laboratory for ubiquitous computing research," in *Cooperative buildings.* Springer, 1999, pp. 191–198.
[16] A. Brush, B. Lee, R. Mahajan, S. Agarwal, S. Saroiu, and C. Dixon, "Home automation in the wild: challenges and opportunities," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2011, pp. 2115–2124.
[17] U. Park and J. Heidemann, "Data muling with mobile phones for sensornets," in *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*, 2011, pp. 162–175.
[18] Bluetooth Core Specification 4.2, 2014. [Online]. Available: https://www.bluetooth.org/en-us/specification/adopted-specifications
[19] Nordic Semiconductor, "IPv6 over Bluetooth Smart," http://www.nordicsemi.com/eng/News/News-releases/Product-Related-News/Nordic-Semiconductor-IPv6-over-Bluetooth-Smart-protocol-stack-for-nRF51-Series-SoCs-enables-small-low-cost-ultra-low-power-Internet-of-Things-applications, 2014.
[20] A. Madhavapeddy, D. Scott, and R. Sharp, "Context-aware computing with sound," in *UbiComp 2003: Ubiquitous Computing*. Springer, 2003, pp. 315–332.
[21] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The cricket location-support system," in *Proceedings of the 6th annual international conference on Mobile computing and networking*. ACM, 2000, pp. 32–43.
[22] G. Borriello, A. Liu, T. Offer, C. Palistrant, and R. Sharp, "Walrus: wireless acoustic location with room-level resolution using ultrasound," in *Proceedings of the 3rd international conference on Mobile systems, applications, and services*. ACM, 2005, pp. 191–203.
[23] P. Lazik and A. Rowe, "Indoor pseudo-ranging of mobile devices using ultrasonic chirps," in *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*. ACM, 2012, pp. 99–112.
[24] A. Madhavapeddy, R. Sharp, D. Scott, and A. Tse, "Audio networking: the forgotten wireless technology," *Pervasive Computing, IEEE*, vol. 4, no. 3, pp. 55–60, 2005.
[25] V. Gerasimov and W. Bender, "Things that talk: using sound for device-to-device and device-to-human communication," *IBM Systems Journal*, vol. 39, no. 3.4, pp. 530–546, 2000.
[26] R. Nandakumar, K. K. Chintalapudi, V. Padmanabhan, and R. Venkatesan, "Dhwani: secure peer-to-peer acoustic nfc," in *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4. ACM, 2013, pp. 63–74.
[27] H. Lee, T. H. Kim, J. W. Choi, and S. Choi, "Chirp signal-based aerial acoustic communication for smart devices," in *Proc. of IEEE Conf. on Computer Communications (INFOCOM), Hong Kong SAR, PRC*, 2015.
[28] C. V. Lopes and P. M. Aguiar, "Aerial acoustic communications," in *Applications of Signal Processing to Audio and Acoustics*. IEEE, 2001, pp. 219–222.
[29] H. Merz, T. Hansemann, and C. Hübner, *Building Automation: Communication Systems with EIB/KNX, LON and BACnet*. Springer Science & Business Media, 2009.
[30] S. Dawson-Haggerty, X. Jiang, G. Tolle, J. Ortiz, and D. Culler, "sMAP: a simple measurement and actuation profile for physical information," in *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*. ACM, 2010, pp. 197–210.
[31] Fhem, http://fhem.de, 2015.
[32] OpenHab, http://www.openhab.org, 2015.
[33] C. J. Plack, *The sense of hearing*. Psychology Press, 2013.
[34] B. C. Moore, *An introduction to the psychology of hearing*. Brill, 2012.
[35] A. J. Berni and W. D. Gregg, "On the utility of chirp modulation for digital signaling," *Communications, IEEE Transactions on*, vol. 21, no. 6, pp. 748–751, 1973.
[36] Q. Dong and W. Dargie, "Evaluation of the reliability of rssi for indoor localization," in *Wireless Communications in Unusual and Confined Areas (ICWCUCA), 2012 International Conference on*. IEEE, 2012, pp. 1–6.
[37] B. Shneiderman, *Designing the user interface-strategies for effective human-computer interaction*. Pearson, 1986.